TASK

Using the attr_data.db, extract all the data from the only table in the database using SQL. Once the data is extracted, import the data into Pandas and continue with the analysis. The following questions should be answered:

1. What do you think are the 3 factors behind employee attrition?
2. What is the relationship between Education and Monthly Income?
3. What is the effect of age on attrition?
4. Is Income the main factor towards employee attrition?
5. How does work-life balance impact the overall attrition rate?

Data Gathering

In [34]:

```
# Import pandas library
import pandas as pd
```

In [35]:

```
# Load your data and print out a few lines. Perform operations to inspect data
# types and look for instances of missing or possibly errant data.
df = pd.read_csv('Stutern_data.csv', sep = ',')
df.head()
```

Out[35]:

	1	41	Yes	Travel_Rarely	1102	Sales	1.1	2	Life Sciences	1.2	...	1.4	80	0	8.1	0.1	1.5	6	4.1	0.2	5
0	2	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	...	4	80	1	10	3	3	10	7	1	7
1	3	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	...	2	80	0	7	3	3	0	0	0	0
2	4	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	...	3	80	0	8	3	3	8	7	3	0
3	5	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	...	4	80	1	6	3	3	2	2	2	2
4	6	32	No	Travel_Frequently	1005	Research & Development	2	2	Life Sciences	1	...	3	80	0	8	2	2	7	7	3	6

5 rows × 36 columns

In [36]:

```
# To get the number of rows and columns of the dataset
df.shape
```

Out[36]:

```
(1469, 36)
```

Notice that the data above has no column name, and it has 1469 records/rows, which is 1 row less because the row with index number 0 represents the header/column title. So I will need add all the colum names from the raw data informaion given

In [37]:

```
# To get all the rows, add the column names while reading csv
df = pd.read_csv('Stutern_data.csv', sep = ',', names=['id','Age','Attrition','BusinessTravel','DailyRate','Department','DistanceFrom
df.head()
```

Out[37]:

	id	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	...	RelationshipSatisfaction
0	1	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	...	1
1	2	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	...	4
2	3	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	...	2
3	4	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	...	3
4	5	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	...	4

5 rows × 36 columns

In [38]:

```
# Now, lets confirm the number of row which we expect to be increased by 1
df.shape
```

Out[38]:

```
(1470, 36)
```

Bravo! Our expected rows of 1470 is now complete

Data Quality Check and Cleaning

In [39]:

```
# Check for duplication
sum(df.duplicated())
```

Out[39]:

```
0
```

In [170]:

```
# Confirm all the columns are correct
df.columns
```

Out[170]:

```
Index(['id', 'Age', 'Attrition', 'BusinessTravel', 'DailyRate', 'Department',
       'DistanceFromHome', 'Education', 'EducationField', 'EmployeeCount',
       'EmployeeNumber', 'EnvironmentSatisfaction', 'Gender', 'HourlyRate',
       'JobInvolvement', 'JobLevel', 'JobRole', 'JobSatisfaction',
       'MaritalStatus', 'MonthlyIncome', 'MonthlyRate', 'NumCompaniesWorked',
       'Over18', 'OverTime', 'PercentSalaryHike', 'PerformanceRating',
       'RelationshipSatisfaction', 'StandardHours', 'StockOptionLevel',
       'TotalWorkingYears', 'TrainingTimesLastYear', 'WorkLifeBalance',
       'YearsAtCompany', 'YearsInCurrentRole', 'YearsSinceLastPromotion',
       'YearsWithCurrManager', 'age_range'],
      dtype='object')
```

In [41]:

```
# data datatype information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 36 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   id                        1470 non-null   int64 
 1   Age                       1470 non-null   int64 
 2   Attrition                 1470 non-null   object
 3   BusinessTravel            1470 non-null   object
 4   DailyRate                 1470 non-null   int64 
 5   Department                1470 non-null   object
 6   DistanceFromHome          1470 non-null   int64 
 7   Education                 1470 non-null   int64 
 8   EducationField            1470 non-null   object
 9   EmployeeCount             1470 non-null   int64 
 10  EmployeeNumber            1470 non-null   int64 
 11  EnvironmentSatisfaction   1470 non-null   int64 
 12  Gender                    1470 non-null   object
 13  HourlyRate                1470 non-null   int64 
 14  JobInvolvement            1470 non-null   int64 
 15  JobLevel                  1470 non-null   int64 
 16  JobRole                   1470 non-null   object
 17  JobSatisfaction           1470 non-null   int64 
 18  MaritalStatus             1470 non-null   object
 19  MonthlyIncome             1470 non-null   int64 
 20  MonthlyRate               1470 non-null   int64 
 21  NumCompaniesWorked        1470 non-null   int64 
 22  Over18                    1470 non-null   object
 23  OverTime                  1470 non-null   object
 24  PercentSalaryHike         1470 non-null   int64 
 25  PerformanceRating         1470 non-null   int64 
 26  RelationshipSatisfaction  1470 non-null   int64 
 27  StandardHours             1470 non-null   int64 
 28  StockOptionLevel          1470 non-null   int64 
 29  TotalWorkingYears         1470 non-null   int64 
 30  TrainingTimesLastYear     1470 non-null   int64 
 31  WorkLifeBalance           1470 non-null   int64 
 32  YearsAtCompany            1470 non-null   int64 
 33  YearsInCurrentRole        1470 non-null   int64 
 34  YearsSinceLastPromotion   1470 non-null   int64 
 35  YearsWithCurrManager      1470 non-null   int64 
dtypes: int64(27), object(9)
memory usage: 413.6+ KB
```

In [44]:

```
# Check for unique values in all columns
# Starting with object types

print ('Attrition-----', df.Attrition.unique())
print ('Business Travel-----', df.BusinessTravel.unique())
print ('Department-----', df.Department.unique())
print ('Education Field-----', df.EducationField.unique())
print ('Gender-----', df.Gender.unique())
print ('Job Role-----', df.JobRole.unique())
print ('Marital Status-----', df.MaritalStatus.unique())
print ('Over 18-----', df.Over18.unique())
print ('Over Time-----', df.OverTime.unique())
```

```
Attrition----- ['Yes' 'No']
Business Travel----- ['Travel_Rarely' 'Travel_Frequently' 'Non-Travel']
Department----- ['Sales' 'Research & Development' 'Human Resources']
Education Field----- ['Life Sciences' 'Other' 'Medical' 'Marketing' 'Technical Degree'
 'Human Resources']
Gender----- ['Female' 'Male']
Job Role----- ['Sales Executive' 'Research Scientist' 'Laboratory Technician'
 'Manufacturing Director' 'Healthcare Representative' 'Manager'
 'Sales Representative' 'Research Director' 'Human Resources']
Marital Status----- ['Single' 'Married' 'Divorced']
Over 18----- ['Y']
Over Time----- ['Yes' 'No']
```

In [95]:

```
df.DistanceFromHome.value_counts()
```

Out[95]:

```
2     211
1     208
10     86
9      85
3      84
7      84
8      80
5      65
4      64
6      59
16     32
11     29
24     28
23     27
29     27
15     26
18     26
26     25
25     25
20     25
28     23
19     22
14     21
12     20
17     20
22     19
13     19
21     18
27     12
Name: DistanceFromHome, dtype: int64
```

Data Visualization

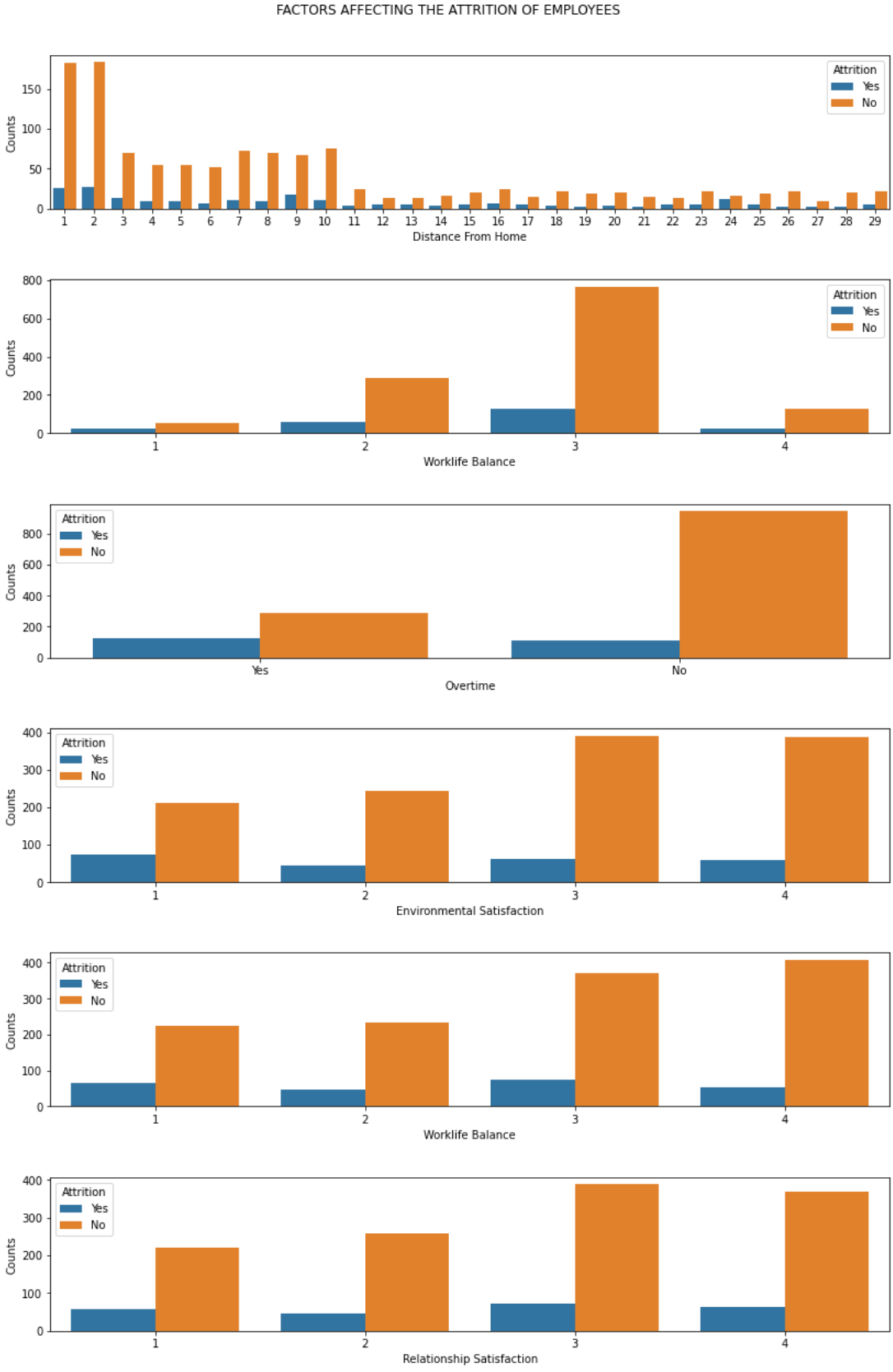
In [90]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

Question 1: What do you think are the 3 factors behind employee attrition?

In [152]:

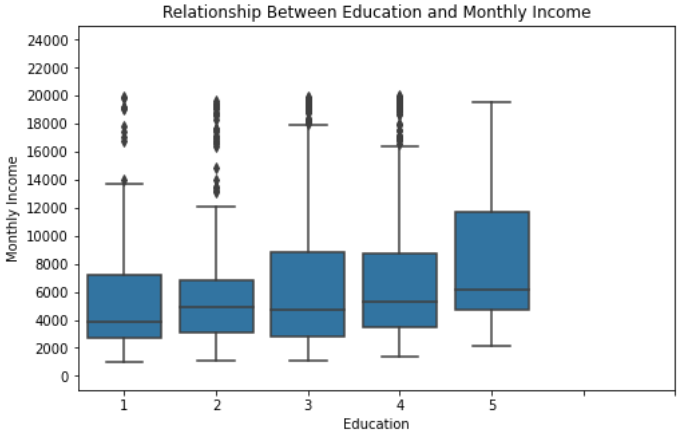
```
# Plot six plots of Attrition versus Factors
fig = plt.figure(figsize=[12,18])
fig.suptitle('FACTORS AFFECTING THE ATTRITION OF EMPLOYEES')

# Subplot 1: distance from home vs Attrition
plt.subplot(6,1,1)
sb.countplot(data = df, x = 'DistanceFromHome', hue = 'Attrition')
plt.ylabel('Counts')
plt.xlabel('Distance From Home')

# Subplot 2: overtime vs Attrition
ax = plt.subplot(6,1,3)
sb.countplot(data = df, x = 'OverTime', hue = 'Attrition')
#plt.xticks(months, month_names)
plt.ylabel('Counts')
plt.xlabel('Overtime')

# Subplot 3: worklife balance vs Attrition
ax = plt.subplot(6,1,2)
sb.countplot(data = df, x = 'WorkLifeBalance', hue = 'Attrition')
# plt.xticks(days, day_names)
plt.ylabel('Counts')
plt.xlabel('Worklife Balance')


# Subplot 4: Environment Satisfaction vs Attrition
ax = plt.subplot(6,1,4)
sb.countplot(data = df, x = 'EnvironmentSatisfaction', hue = 'Attrition')
# plt.xticks(days, day_names)
plt.ylabel('Counts')
plt.xlabel('Environmental Satisfaction')

# Subplot 5: Job Satisfaction vs Attrition
ax = plt.subplot(6,1,5)
sb.countplot(data = df, x = 'JobSatisfaction', hue = 'Attrition')
# plt.xticks(days, day_names)
plt.ylabel('Counts')
plt.xlabel('Worklife Balance')

# Subplot 6: Relationship Satisfaction vs Attrition
ax = plt.subplot(6,1,6)
sb.countplot(data = df, x = 'RelationshipSatisfaction', hue = 'Attrition')
# plt.xticks(days, day_names)
plt.ylabel('Counts')
plt.xlabel('Relationship Satisfaction')

fig.tight_layout(pad = 3.0)
plt.show()
```


FACTORS AFFECTING THE ATTRITION OF EMPLOYEES
Attrition
Yes
No
Counts
Distance From Home
Worklife Balance
Overtime
Environmental Satisfaction
Worklife Balance
Relationship Satisfaction

Question 2: What is the relationship between Education and Monthly Income?

In [138]:

```
# set figure size
plt.figure(figsize=[8, 5])

base_color = sb.color_palette()[0]

sb.boxplot(data = df, x = 'Education', y = 'MonthlyIncome',color=base_color)

plt.xticks(range(7), rotation=0)
plt.yticks(np.arange(0, 30000+1, 2000))

plt.ylim(-1000,25000)

# labels and title
plt.xlabel('Education')
plt.ylabel('Monthly Income')
plt.title('Relationship Between Education and Monthly Income');
```



We can see that the average income for increases with higher degree of education

Question 3: What is the effect of age on attrition?

I intend to group these ages into 3:

- children: age 0-18 years
- young_adults: age 18-40 years
- older_adults: age 40-60 years

In [117]:

```
bin_edges = [ 0, 18, 40, 60]
bin_names = ['children', 'young_adults', 'older_adults']
# Create age_range column
df['age_range'] = pd.cut(df['Age'], bin_edges, labels=bin_names)
# Checks for successful creation of this column
df['age_range'].value_counts()
```

Out[117]:

```
young_adults    997
older_adults    465
children          8
Name: age_range, dtype: int64
```

In [119]:

```
age_distribution= df.groupby('age_range')['Attrition'].value_counts().unstack()
print(age_distribution)
print(age_distribution.sum())
```

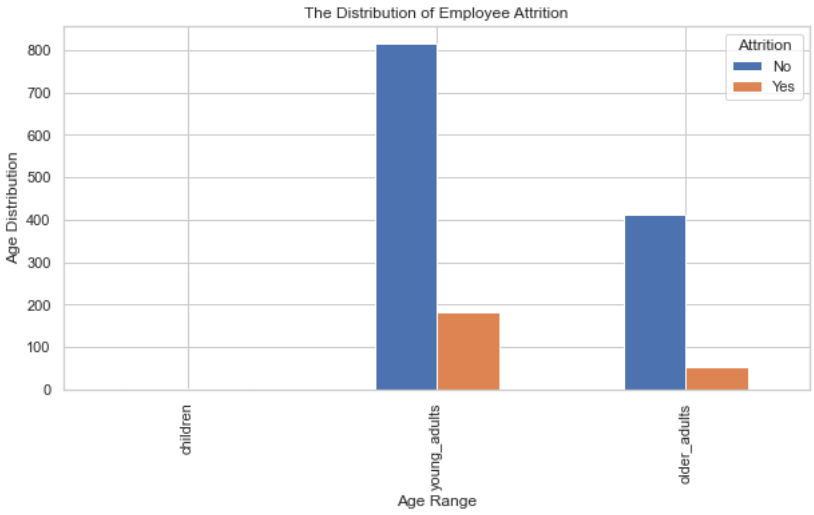
```
Attrition      No  Yes
age_range
children        4    4
young_adults  816  181
older_adults  413   52
Attrition
No     1233
Yes     237
dtype: int64
```

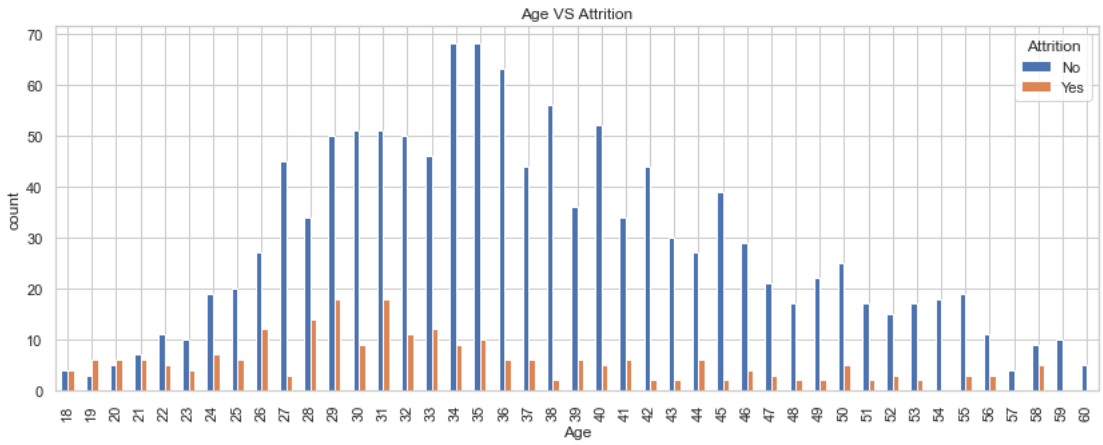
In [185]:

```
age_distribution.plot(kind='bar', figsize=(10,5), title='The Distribution of Employee Attrition')
plt.xlabel('Age Range', fontsize=12)
plt.ylabel('Age Distribution', fontsize=12)


attrition_effect= df.groupby('Age')['Attrition'].value_counts().unstack()
attrition_effect.plot(kind='bar', figsize=(14,5), title='Age VS Attrition');
plt.xlabel('Age', fontsize=12)
plt.ylabel('count', fontsize=12)
```

Out[185]:

```
Text(0, 0.5, 'count')
```





Overall, the young adults of age 18-40years have the largest population, with majority of them remaining in the job. The population of employee attrition in young adults is seen to be more than those above 40years.

While the younger staffs have the largest attrition, the younger employees between 26-35years left their job the most, with those aged 29years and 31years being the top on the list.

Question 4: Is Income the main factor towards employee attrition?

Yes

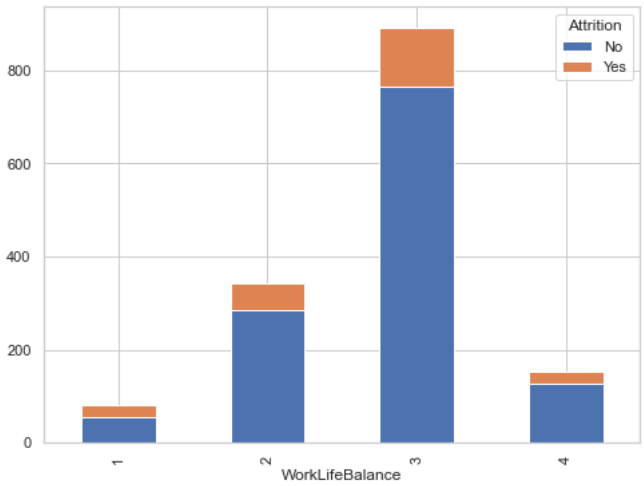
Question 5: How does work-life balance impact the overall attrition rate?

In [181]:

```
1 df.groupby(['WorkLifeBalance', 'Attrition']).size().unstack().plot(kind='bar', stacked=True, figsize=(8, 6))
```

Out[181]:

```
<AxesSubplot:xlabel='WorkLifeBalance'>
```



The output below shows that the attrition rate is the highest among employees with work life balance in level 3